A. REMARKS

The Examiner is thanked for the performance of another search. No amendments have been made herein. Hence, Claims 1-4, 6-9, 16-23, 25-28 and 30-32 are pending in this application. All issues raised in the Office Action mailed March 10, 2004 are addressed hereinafter.

REJECTION OF CLAIMS 1-4, 6-9, 16-23, 25-28 AND 30-32 UNDER 35 U.S.C. § 103(a)

Claims 1-4, 6-9, 16-23, 25-28 and 30-32 were rejected under 35 U.S.C. § 103(a) as being unpatentable over *Aahlad et al.*, U.S. Patent No. 5,969,967 (hereinafter "*Aahlad*") in view of *Misheski et al.*, U.S. Patent No. 5,915,252 (hereinafter "*Misheski*"). It is respectfully submitted that Claims 1-4, 6-9, 16-23, 25-28 and 30-32 are patentable over *Aahlad* and *Misheski* for at least the reasons provided hereinafter.

Aahlad describes an apparatus for providing conspiracy among objects. As used in Aahald, the term "conspiracy" refers to the ability for objects to communicate with each other "behind" object interfaces and share resources. *Aahlad* describes three approaches for providing conspiracy among objects: (1) a shared servant conspiracy model; (2) a shared object conspiracy model; and (3) a hybrid object conspiracy model.

In the shared servant conspiracy model, each of the conspirator objects resides in a common server process and has a pointer to a conspiracy servant object also resident in the server process. When a client calls one of the conspirator objects to invoke an object operation, the conspirator skeleton, having the direct pointer, passes the invocation from the client directly to the conspiracy servant object. The requested service is then performed by the conspiracy servant object and the results returned to the client via the object resource broker (ORB). Thus

in the shared servant conspiracy model, the conspirator objects conspire by way of a pointer and must all reside in the same process (Col. 10, lines 31-54).

In the shared object conspiracy model, each conspirator object has a pointer to its own conspirator servant. All conspirator servants have a pointer to a shared conspiracy servant, i.e., the shared object. In operation, when a client calls one of the conspirator objects to invoke an object operation, the corresponding conspirator skeleton for the conspirator object invokes the operation on the conspirator servant for the conspirator object. The conspirator servant invokes the operation of the conspiracy skeleton, which invokes the operation on the (shared) conspiracy servant. The conspiracy servant object performs the requested service and passes any results of the requested operation back out to the client (Col. 12, lines 39-59).

The hybrid conspiracy model is an extension of the shared object conspiracy model. Like in the shared object conspiracy model, each conspirator object has a pointer to its own conspirator servant. All conspirator servants have a pointer to a shared conspiracy servant. In the hybrid conspiracy model, the shared conspiracy servant has a pointer to a conspiracy engine. Upon activation of a conspirator object, the activated conspirator object invokes a request for the pointer to the conspiracy engine on the conspiracy object reference. In response, the conspiracy servant passes the pointer to the conspiracy engine back to the conspirator servant. Then, upon receiving client requests, the conspirator servant passes the service request directly to the conspiracy engine for processing (Col. 14, line 48 through Col. 15, line 3).

CLAIM 1

Claim 1 recites a method for processing data on a distributed computing system that includes a plurality of nodes that requires the steps of:

> "maintaining mapping data that specifies work that can be performed by each of the plurality of nodes;
> in response to receiving a first work request to perform first work from a first process on a first node from the plurality of nodes, determining based upon the first work and the mapping data, that the first work is to be performed on a second node from the plurality of nodes;
> providing the first work request to a second process on the second node, wherein the first work request specifies that the first process is to directly receive results of the first work;
> determining based upon the first work and the mapping data, that the first work is also to be performed on a third node from the plurality of nodes, and
> providing a second work request to a third process on the third node, wherein the second work request specifies that results of the first work performed on the third node are to be provided directly to the first process."

It is respectfully submitted that Claim 1 includes at least one limitation that is not in any way taught or suggested by *Aahlad* and *Misheski*, considered alone or in combination. In *Aahlad*, none of the three described conspiracy models, i.e., the shared servant conspiracy model, the shared object conspiracy model or the hybrid object conspiracy model, include determining, based upon a received work request to perform first work, that the first work is to be performed by two different processes on two different nodes, i.e., both the second and third processes on the second and third nodes, and then providing first and second work requests to the second and third processes.

In all three conspiracy models of *Aahlad*, each client call results in the invocation of a single process on a single node. In the shared servant and shared object conspiracy models, each client call results in the invocation of a single conspiracy servant. In the hybrid conspiracy model, each client call results in the invocation of a single conspiracy engine. In none of these

models are two separate work requests provided to two separate processes on two separate nodes for processing in response to receipt of a single work request, as is required by Claim 1. It is therefore respectfully submitted that the limitations of:

> in response to receiving a first work request to perform first work from a first process on a first node from the plurality of nodes, determining based upon the first work and the mapping data, that the first work is to be performed on a second node from the plurality of nodes;
> providing the first work request to a second process on the second node, wherein the first work request specifies that the first process is to directly receive results of the first work;
> determining based upon the first work and the mapping data, that the first work is also to be performed on a third node from the plurality of nodes, and
> providing a second work request to a third process on the third node, wherein the second work request specifies that results of the first work performed on the third node are to be provided directly to the first process

are not taught or suggested by *Aahlad.*

Misheski was relied upon in the Office Action for teaching the "maintaining mapping data" limitation recited in Claim 1. It is respectfully submitted that *Misheski* does not in any way teach or suggest this limitation. *Misheski* describes an object-oriented framework for facilitating data transfer between a data source and a data target. It is respectfully submitted that there is no teaching or suggestion in *Misheski* of "maintaining mapping data that specifies work that can be performed by each of the plurality of nodes," as required by Claim 1 and that the portions of *Misheski* referred to in the Office Action do not in any way teach or suggest this limitation. For example, the text at Col. 12, lines 40-53 describes a conventional computer system architecture. The text at Col. 13, lines 52-65 describes how data may be transferred between any data source and data target over different types of communication media. The text at Col. 14, lines 16-31 describes how, prior to transfer to a data target, data on a data source may be mapped to the data target. *Misheski* describes that the mapping includes converting file formats from a format

understood by the data source to a format understood by the data target. Thus, it is respectfully submitted that to the extent this portion of *Misheski* teaches or suggests mapping data, it is only in the context of data that specifies how data is to be converted between different formats prior to the data being transferred to the data target. The text at Col. 15, lines 12-50 describes a class diagram used to implement the data transfer network. This includes a description of a Target Mapping class used to reformat data from a data source to a data target. The text at Col. 18, lines 12-43 describes how the map() method may be invoked to provide data translation between different formats. The text at Col. 26, lines 25-67 is a Claim 15 that recites a method for performing data transfer from a data source to a data target that includes providing a target mapping object used in an object-oriented framework mechanism to transfer the data from the data source to the data target. Based upon the foregoing, it is respectfully submitted that none of these portions of *Misheski* teach or suggest mapping data as recited in Claim 1, i.e., mapping data "that specifies work that can be performed by each of the plurality of nodes."

In view of the foregoing, it is respectfully submitted that Claim 1 includes at least one limitation that is not taught or suggested by *Aahlad* and *Misheski*, alone or in combination and is therefore patentable over *Aahlad* and *Misheski*.

CLAIMS 2-4 AND 6-9

Claims 2-4 and 6-9 depend from Claim 1 and include all of the limitations of Claim 1. It is therefore respectfully submitted that Claims 2-4 and 6-9 are patentable over *Aahlad* and *Misheski* for at least the reasons set forth herein with respect to Claim 1. Furthermore, it is respectfully submitted that Claims 2-4 and 6-9 recite additional limitations that independently render them patentable over *Aahlad* and *Misheski*.

CLAIMS 16-19

Claims 16-19 include limitations similar to Claims 1-4 and 6-9, except in the context of a distributed computing system. It is therefore respectfully submitted that Claims 16-19 are patentable over *Aahlad* and *Misheski* for at least the reasons set forth herein with respect to Claims 1-4 and 6-9.

CLAIMS 20-23 AND 25-28

Claims 20-23 and 25-28 include limitations similar to Claims 1-4 and 6-9, except in the context of a computer-readable medium. It is therefore respectfully submitted that Claims 20-23 and 25-28 are patentable over *Aahlad* and *Misheski* for at least the reasons set forth herein with respect to Claims 1-4 and 6-9.

CLAIMS 30-32

It is respectfully submitted that Claim 31 includes at least one limitation that is not in any way taught or suggested by *Aahlad* and *Misheski*, alone or in combination. For example, Claim 31 requires the step of "generating an updated first work request that specifies that the first process is to directly receive results of performing the first work" and "providing the updated first work request to a second process on the second node." It is respectfully submitted that none of the three conspiracy models described in *Aahlad*, i.e., the shared servant conspiracy model, the shared object conspiracy model or the hybrid object conspiracy model, include generating updated work requests. In all three models, client calls cause the invocation of an object via addressing. There is no mention or suggestion of generating an updated work request and providing the updated work request to, for example, the conspiracy servant in the shared servant and shared object conspiracy models, or the conspiracy engine in the hybrid conspiracy model.

The Office Action refers to the text at Col. 10, lines 31-54 of *Aahlad* in support of this rejection. This portion of *Aahlad* describes the shared servant conspiracy model. More specifically, this portion of *Aahlad* describes how when a client calls one of the conspirator objects to invoke an object operation, the conspirator skeleton, having the direct pointer, passes the invocation from the client directly to the conspiracy servant object. The requested service is then performed by the conspiracy servant object and the results returned to the client via the object resource broker (ORB). There is no mention in this, or any other portion, of Aahald of generating an updated work request as recited in Claim 31. Based on the foregoing, it is respectfully submitted that Claim 31 includes at least one limitation that is not in any way taught or suggested by *Aahlad* and *Misheski*, alone or in combination, and is therefore patentable over *Aahlad* and *Misheski*.

Claims 30 and 32 recite limitations similar to Claim 31, except in the context of a method and computer-readable medium, respectively.

In view of the foregoing, reconsideration and withdrawal of the rejection of Claims 1-4, 6-9, 16-23, 25-28 and 30-32 under 35 U.S.C. § 103(a) as being unpatentable over *Aahlad* in view of *Misheski* is respectfully requested.

It is respectfully submitted that all of the pending claims are in condition for allowance and the issuance of a notice of allowance is respectfully requested. If there are any additional charges, please charge them to Deposit Account No. 50-1302.

The Examiner.is invited to contact the undersigned by telephone if the Examiner believes that such contact would be helpful in furthering the prosecution of this application.

Respectfully submitted,

HICKMAN PALERMO TRUONG & BECKER LLP

Dated: June 7, 2004

Edward A. Becker
Reg. No. 37,777

1600 Willow Street
San Jose, CA 95125
(408) 414-1204
Facsimile: (408) 414-1076

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: **Mail Stop Amendment**, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450

On June 7, 2004 By

Angelica Jaquez